Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Canceled)

2. (Currently Amended) ~~The logic circuit of claim 1 further comprising:~~ A logic circuit comprising:

a first look-up table (LUT) in a first logic element (LE), wherein the first LUT produces a carry from a first set of corresponding bits of at least three binary numbers; and

a second LUT in a second LE, wherein the second LUT produces a sum from a second set of corresponding bits of the binary numbers;

an adder in the second LE, wherein the adder is coupled with the first LUT and the second LUT, and wherein the adder adds the carry and the sum; and

a multiplexer in the second LE, wherein the multiplexer is coupled to the first LUT and a third LUT in the second LE, further wherein the multiplexer selects between a signal received from the first LUT and a signal received from the third LUT for forwarding to the adder.

3. (Currently Amended) The logic circuit of claim ~~1~~2, wherein the adder comprises a hardwired adder.

4. (Currently Amended) The logic circuit of claim ~~1~~2, wherein the first LUT determines a 1 bit carry of three binary numbers in a carry save adder process.

5. (Currently Amended) The logic circuit of claim ~~1~~2, wherein the first LUT receives bits X[n], Y[n], and Z[n] and provides as an output ((X[n] AND Y[n]) OR (X[n] AND

Z[n]) OR (Y[n] AND Z[n])), wherein n is an integer, X[n], Y[n], and Z[n] are the n-th bits of binary numbers X, Y, and Z, further wherein OR designates the Boolean OR function, and AND designates the Boolean AND function.

6. (Currently Amended) The logic circuit of claim ~~1~~2, wherein the second LUT determines a 1 bit sum of three binary numbers in a carry save adder process.

7. (Currently Amended) The logic circuit of claim ~~1~~2, wherein the second LUT receives bits X[n+1], Y[n+1], and Z[n+1] and provides as an output (X[n+1] XOR Y[n+1]) XOR Z[n+1]), wherein n is an integer, X[n+1], Y[n+1], and Z[n+1] are the (n+1)-th bits of binary numbers X, Y, and Z, further wherein XOR designates the Boolean XOR (exclusive OR) function.

8. (Currently Amended) A programmable logic device including the logic circuit of claim ~~1~~2.

9. (Currently Amended) A digital system comprising a programmable logic device including the logic circuit of claim ~~1~~2.

10. (Previously Presented) A logic element (LE) comprising:

a multiplexer; and

an adder coupled to the multiplexer,

wherein the multiplexer selects between a signal determined in the LE and a signal determined in a previous LE for forwarding to the adder, and wherein the multiplexer selects the signal determined in the previous LE when the LE is set to operate in an addition of three binary numbers mode.

11. (Original) The LE of claim 10, wherein the adder comprises a hardwired adder.

12. (Original) The LE of claim 10 further comprising a first logic circuit that determines a sum of an (n+1)-th bit of three binary numbers in a carry save adder process, wherein n is an integer.

13. (Original) The LE of claim 12, wherein the logic circuit comprises a look-up table (LUT).

14. (Original) The LE of claim 10, wherein the previous LE comprises a second logic circuit that determines a carry of an n-th bit of three binary numbers in a carry save adder process, wherein n is an integer.

15. (Original) The LE of claim 14, wherein the logic circuit comprises a look-up table (LUT).

16. (Original) A programmable logic device including the LE of claim 10.

17. (Original) A digital system comprising a programmable logic device including the LE of claim 10.

18-25. (Canceled)

26. (Currently Amended) ~~The PLD of claim 25,~~ A programmable logic device (PLD) comprising:

a first logic element (LE), the first LE including:

a first look-up table (LUT), wherein the first LUT determines a first sum;

a second LUT, wherein the second LUT determines a first carry;

a third LUT, wherein the third LUT determines a second sum;

a fourth LUT, wherein the fourth LUT determines a second carry;

a first hardwired adder, wherein the first hardwired adder receives the first sum from the first LUT and a carry from a LUT of a previous LE or ground signals; and

a second hardwired adder, wherein the second hardwired adder receives the first carry and the second sum; and

a second LE, the second LE including:

a fifth LUT, wherein the fifth LUT determines a third sum;

a sixth LUT, wherein the sixth LUT determines a third carry;

a seventh LUT, wherein the seventh LUT determines a fourth sum;

an eighth LUT, wherein the eighth LUT determines a fourth carry;

a third hardwired adder, wherein the third hardwired adder receives the third sum determined by the fifth LUT and the second carry determined by the fourth LUT; and

a fourth hardwired adder, wherein the fourth hardwired adder receives the fourth sum and the third carry,

wherein:

the first LE further comprises:

a first multiplexer having first and second input terminals and a first output terminal, wherein the first input terminal is coupled to one of ground signals or an output of the LUT of the previous LE, the second input terminal is coupled to a first output terminal of the second LUT, and the first output terminal is coupled to the first hardwired adder; and

a second multiplexer having third and fourth input terminals and a second output terminal, wherein the third input terminal is coupled to a second output terminal of the second LUT, the fourth input terminal is coupled to a first output terminal of the fourth LUT, and the second output terminal is coupled to the second hardwired adder; and

the second LE further comprises:

a third multiplexer having fifth and sixth input terminals and a third output terminal, wherein the fifth input terminal is coupled to a second output terminal of the

fourth LUT, the sixth input terminal is coupled to a first output terminal of the sixth LUT, and the third output terminal is coupled to the third hardwired adder; and

a fourth multiplexer having seventh and eighth input terminals and a fourth output terminal, wherein the seventh input terminal is coupled to a second output terminal of the sixth LUT, the fourth input terminal is coupled to a first output terminal of the eighth LUT, and the fourth output terminal. is coupled to the fourth hardwired adder.

27. (Currently Amended) The PLD of claim ~~25~~26, wherein the first sum, the first carry, the second sum, the second carry, the third sum, the third carry, the fourth sum, and the fourth carry, are sums and carrys in a carry save adder process.

28. (Currently Amended) A digital system including the PLD of claim ~~25~~26.

29. (Currently Amended) A method of operating a programmable logic device to add ~~adding~~ at least three binary numbers having multiple bits ~~using a programmable logic device~~, the method comprising:

receiving a first set of corresponding bits of the binary numbers at a first look-up table (LUT) of a first logic element (LE);

producing a carry from the first LUT and sending the carry to a multiplexer;

receiving a second set of corresponding bits of the binary numbers at a second LUT of a second LE;

producing a sum from the second LUT;

selecting, with the multiplexer, between the carry received from the first LUT and a signal received from a third LUT in the second LE for forwarding to an adder, wherein the multiplexer is coupled to the first LUT and the third LUT;

receiving, at ~~an~~ the adder, the carry and the sum;

adding the carry produced by the first LUT of the first LE with the sum produced by ~~a~~ the second LUT of the second LE to obtain at least part of a total sum of the binary numbers; and

outputting the at least part of the total sum of the binary numbers from the adder.

30. (Canceled)

31. (Previously Presented) The method of claim 29, wherein the determining the carry includes determining a 1 bit carry of three binary numbers in a carry save adder process.

32. (Original) The method of claim 31, wherein the determining the carry includes:

receiving bits X[n], Y[n], and Z[n]; and

determining ((X[n] AND Y[n]) OR (X[n] AND Z[n]) OR (Y[n] AND Z[n])), wherein n is an integer, X[n], Y[n], and Z[n] are the n-th bits of binary numbers X, Y, and Z, further wherein OR designates the Boolean OR function, and AND designates the Boolean AND function.

33. (Previously Presented) The method of claim 29, wherein the determining the sum includes determining a 1 bit sum of three binary numbers in a carry save adder process.

34. (Original) The method of claim 33, wherein the determining the sum includes:

receiving bits X[n+1], Y[n+1], and Z[n+1]; and

determining (X[n+1] XOR Y[n+1]) XOR Z[n+1]), wherein n is an integer, X[n+1], Y[n+1], and Z[n+1] are the (n+1)-th bits of binary numbers X, Y, and Z, further wherein XOR designates the Boolean XOR (exclusive OR) function.

35. (Previously Presented) The method of claim 29, further comprising:

receiving, at the adder, a carry over signal from another adder; and

adding the carry over signal to the carry and the sum to obtain the at least part of a total sum of the binary numbers.

36. (Previously Presented) The method of claim 29 wherein producing a carry by the first LUT uses only the first set of corresponding bits.